

# UI тесты мобильных приложений: ожидание и реальность

Устинова Анна, тимлид команды  
тестирования мобильных решений  
DIRECTUM

# О чем

Основные виды автотестов и принципы их применения.

Наш опыт автоматизации:

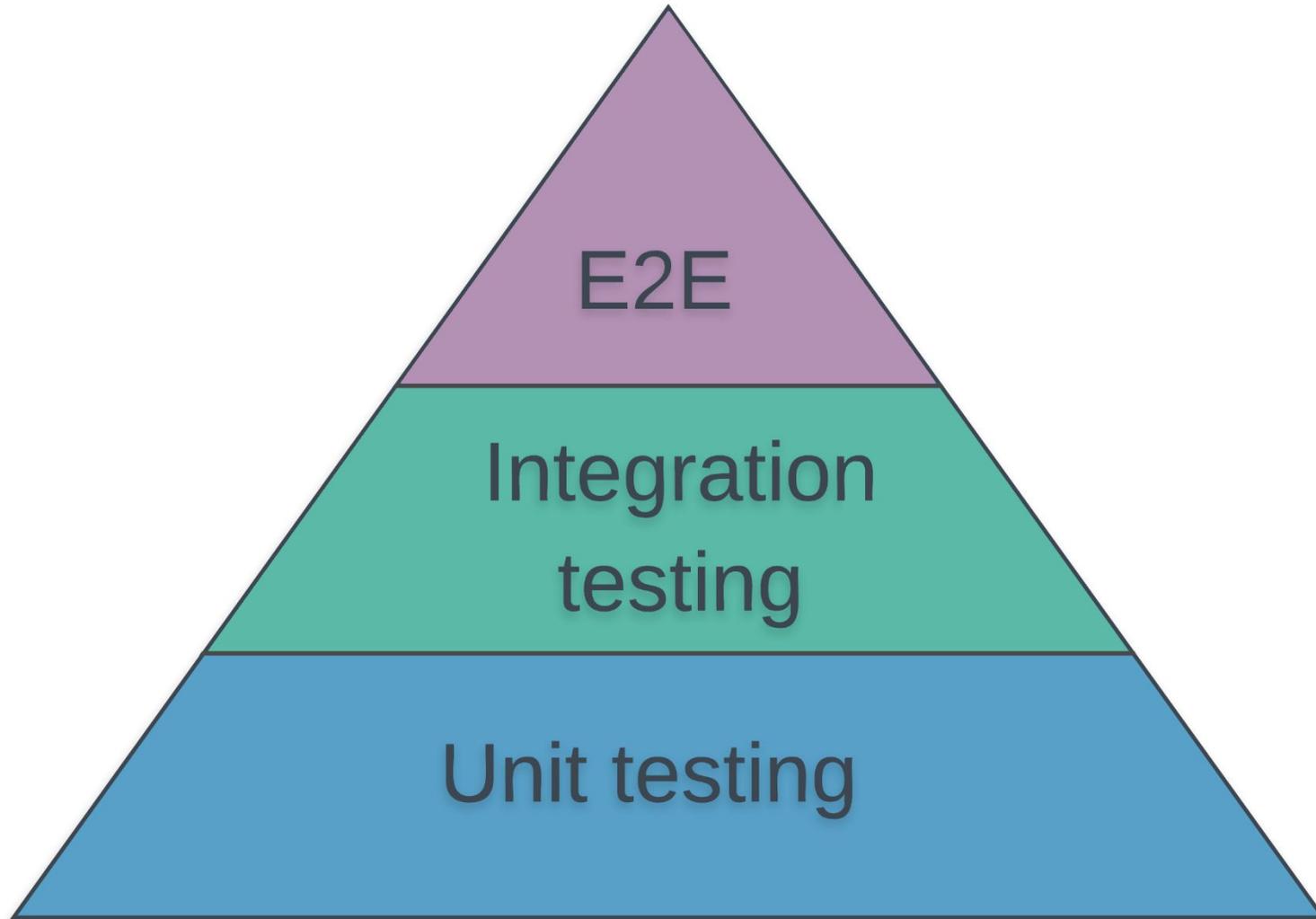
- какие у нас АТ и почему;
- XamarinUI.Test;
- как АТ внедрены в процесс разработки.

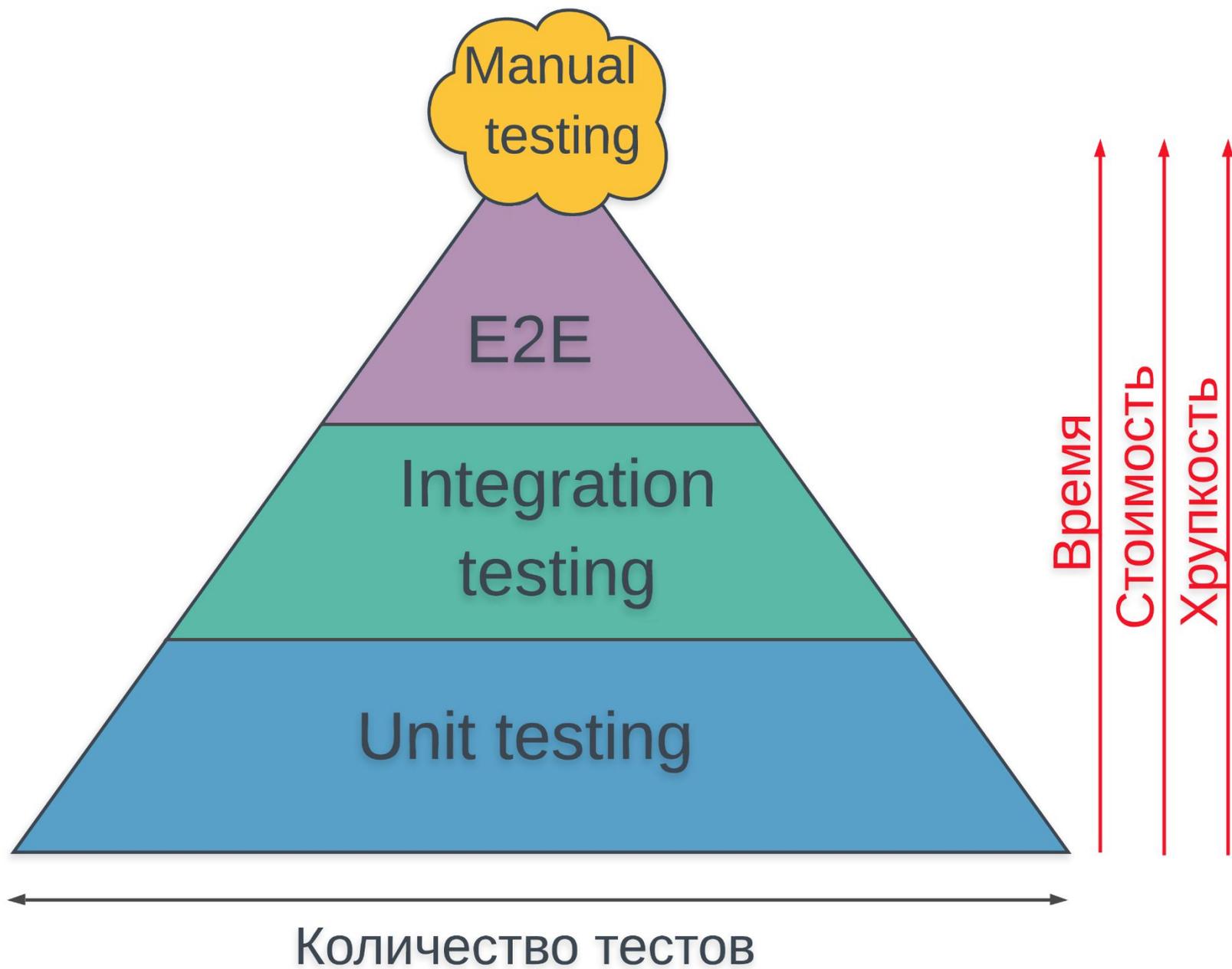
# Продукты

- DIRECTUM, DirectumRX;
- веб-сервер NOMAD;
- мобильные приложения DIRECTUM Solo, DIRECTUM Jazz.

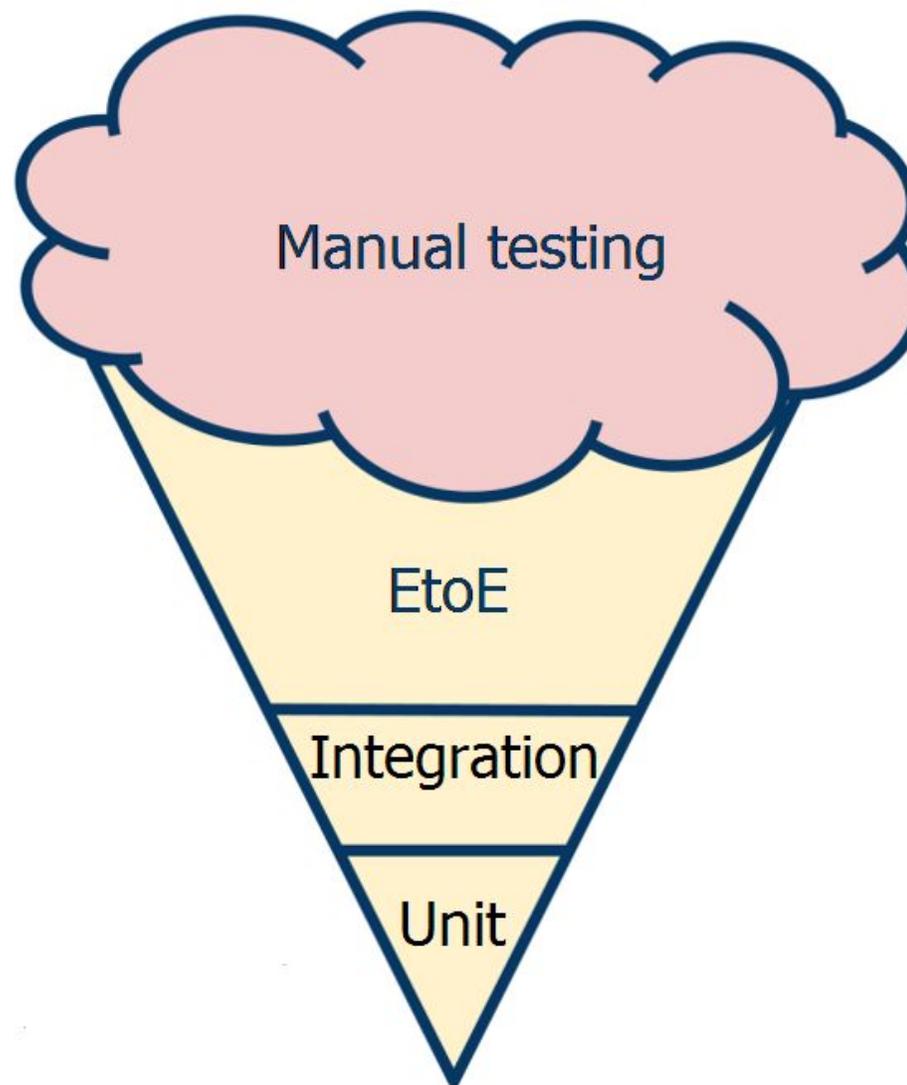


# Пирамида автоматизации тестирования





# Антипаттерн



# Виды тестирования на примере наших продуктов

# Интеграционное

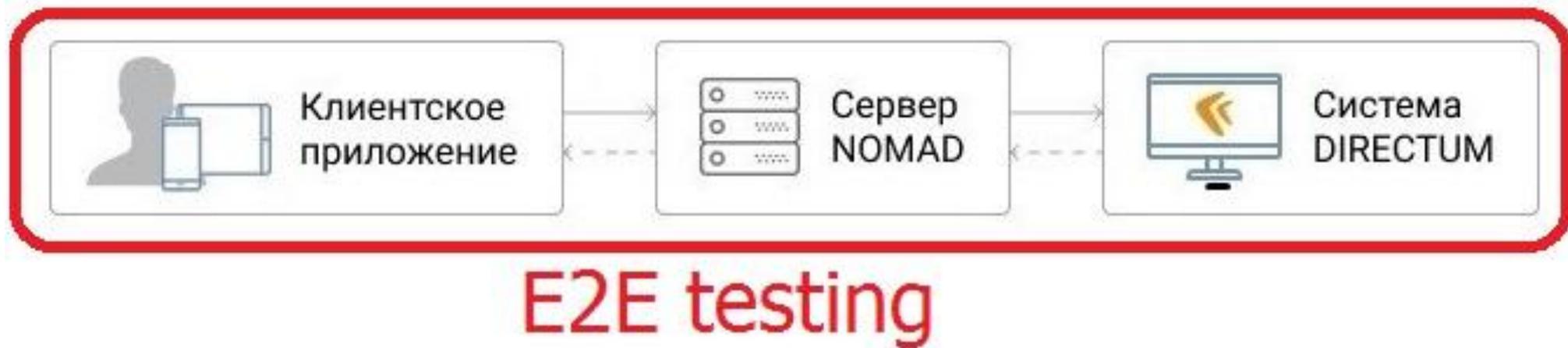


## Integration testing

Проверяют взаимодействие веб-сервиса с СЭД через API.

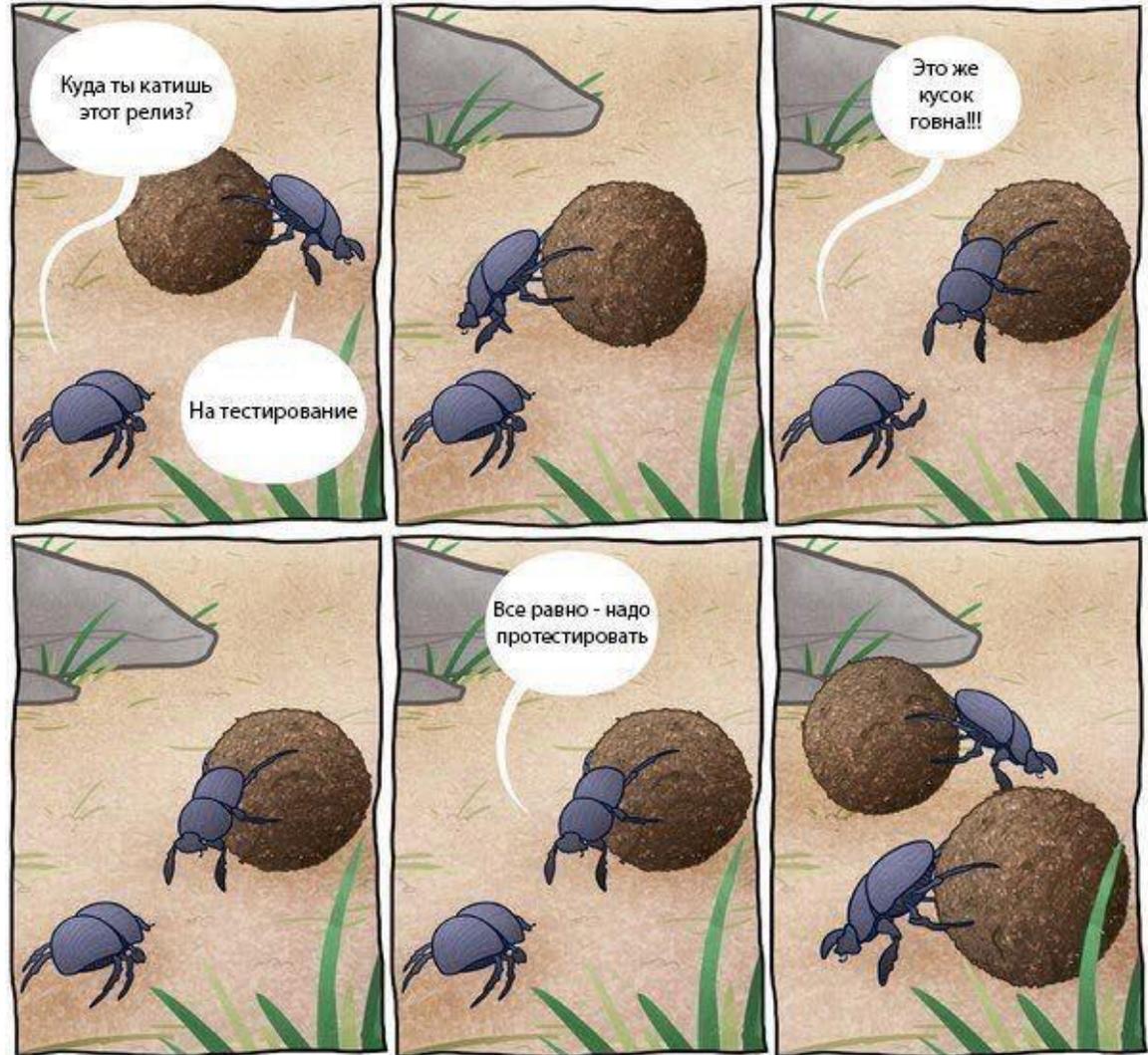
Имитируется работа клиентского приложения.

# Сквозное тестирование



Через UI приложения обращаются к веб-сервису.  
Веб-сервис взаимодействует с СЭД.  
Имитируется полноценная работа конечного пользователя.

# Зачем нам АТ мобильных приложений



# Зачем нам АТ мобильных приложений

Покрыть АТ основные кейсы приложения

Уменьшить количество багов на регрессии

# Текущие цели

Уменьшение количества ручных регрессионных тестов

Раннее нахождение багов в стабильной функциональности

Проверка UI

# Когда пора внедрять АТ

Есть стабильная функциональность

Есть понимание, что приложение будет развиваться дальше

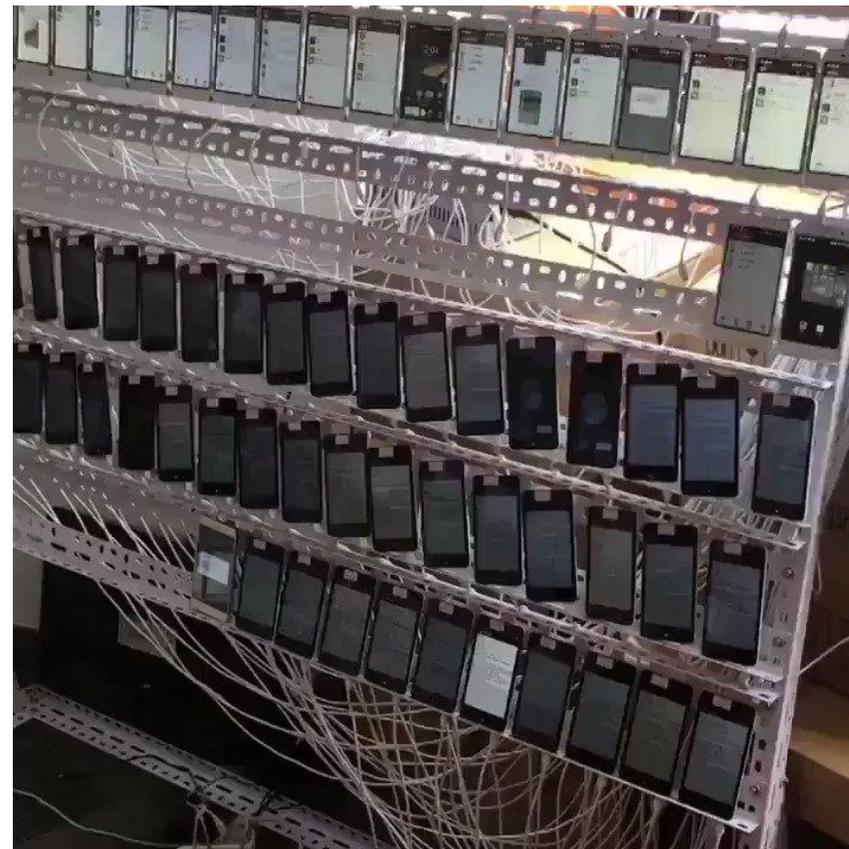
Есть ресурсы (на поддержку и развитие)

# Автотесты мобильных приложений. Ожидания

Тестируют UI

Большое покрытие  
конфигураций

Массмаркет, частые релизы



# Наши автотесты. Реальность

Прогоняются раз в день

Локально

На двух девайсах (iOS, Android)

Минимально проверяют UI, упор на функциональность

Android ~ 40 тестов, ~ 50 минут

iOS ~ 30 тестов, ~ 20 минут

Написаны с использованием Xamarin.UITest

# Xamarin.UITest

- фреймворк для автоматического тестирования UI;
- C#, NUnit;
- Xamarin.iOS, Xamarin.Android проекты (но поддерживает и проекты на Objective-C / Swift и Java).

# Встраивание в проект

iOS:

NuGet пакет Xamarin Test Cloud Agent

в метод AppDelegate.FinishedLaunching:

```
#if ENABLE_TEST_CLOUD
```

```
Xamarin.Calabash.Start();
```

```
#endif
```

Android не требует дополнительной настройки в проекте.

# Принцип работы

Поиск элемента (Queries)

Взаимодействие с ним (Actions):

- Tap;
- Swipe;
- EnterText;
- ..

# Пример теста

```
public void ShowErrIncorrectLoginOrPassword_IfLoginIsWrong()  
{  
    var wrongLogin = TestsSettings.UserLogin + "1";  
    app.EnterLoginAndPassword(wrongLogin,  
TestsSettings.UserPassword);  
    app.WaitForElement(Resources.Identifiers.ErrorMessage, "Login  
is incorrect, alert message wasn't shown.", TestsSettings.Delay);  
    Assert.AreEqual(CoreResources.ErrIncorrectLoginOrPassword,  
ErrorMessage);  
}
```

# Пример теста

```
public void ShowErrIncorrectLoginOrPassword_IfLoginIsWrong()
{
    var wrongLogin = TestsSettings.UserLogin + "1";
    app.EnterLoginAndPassword(wrongLogin,
TestsSettings.UserPassword);
    app.WaitForElement(Resources.Identifiers.ErrorMessage, "Login
is incorrect, alert message wasn't shown.", TestsSettings.Delay);
    Assert.AreEqual(CoreResources.ErrIncorrectLoginOrPassword,
ErrorMessage);
}
```

# Пример теста

```
public void ShowErrIncorrectLoginOrPassword_IfLoginIsWrong()
{
    var wrongLogin = TestsSettings.UserLogin + "1";
    app.EnterLoginAndPassword(wrongLogin,
TestsSettings.UserPassword);
    app.WaitForElement(Resources.Identifiers.ErrorMessage, "Login
is incorrect, alert message wasn't shown.", TestsSettings.Delay);
    Assert.AreEqual(CoreResources.ErrIncorrectLoginOrPassword,
ErrorMessage);
}
```

# Пример теста

```
public void ShowErrIncorrectLoginOrPassword_IfLoginIsWrong()
{
    var wrongLogin = TestsSettings.UserLogin + "1";
    app.EnterLoginAndPassword(wrongLogin,
TestsSettings.UserPassword);
    app.WaitForElement(Resources.Identifiers.ErrorMessage, "Login is
incorrect, alert message wasn't shown.", TestsSettings.Delay);
    Assert.AreEqual(CoreResources.ErrIncorrectLoginOrPassword,
ErrorMessage);
}
```

# Пример теста

```
public void ShowErrIncorrectLoginOrPassword_IfLoginIsWrong()
{
    var wrongLogin = TestsSettings.UserLogin + "1";
    app.EnterLoginAndPassword(wrongLogin, TestsSettings.UserPassword);
    app.WaitForElement(Resources.Identifiers.ErrorMessage, "Login is incorrect,
alert message wasn't shown.", TestsSettings.Delay);
    Assert.AreEqual(CoreResources.ErrIncorrectLoginOrPassword, ErrorMessage);
}
```

```
private string ErrorMessage =>
```

```
app.Query(x =>
x.Marked(Resources.Identifiers.ErrorMessage)).First().Text;
```

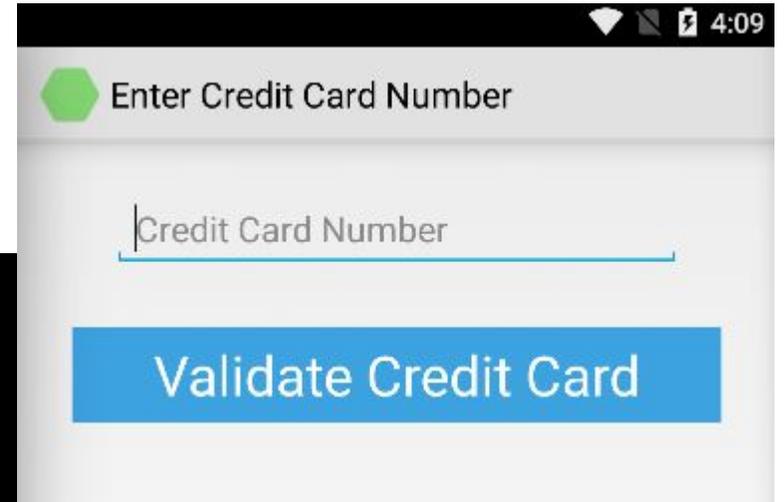
# Пример теста

```
public static void EnterLoginAndPassword(this AndroidApp app,  
string login, string password)  
{  
    app.WaitForElement(Resources.Identifiers.LoginEdit,  
TestsSettings.Delay);  
    app.EnterText(Resources.Identifiers.LoginEdit, login);  
    app.EnterText(Resources.Identifiers.PasswordEdit,  
password);  
    app.Tap(Resources.Identifiers.LoginButton);  
}
```



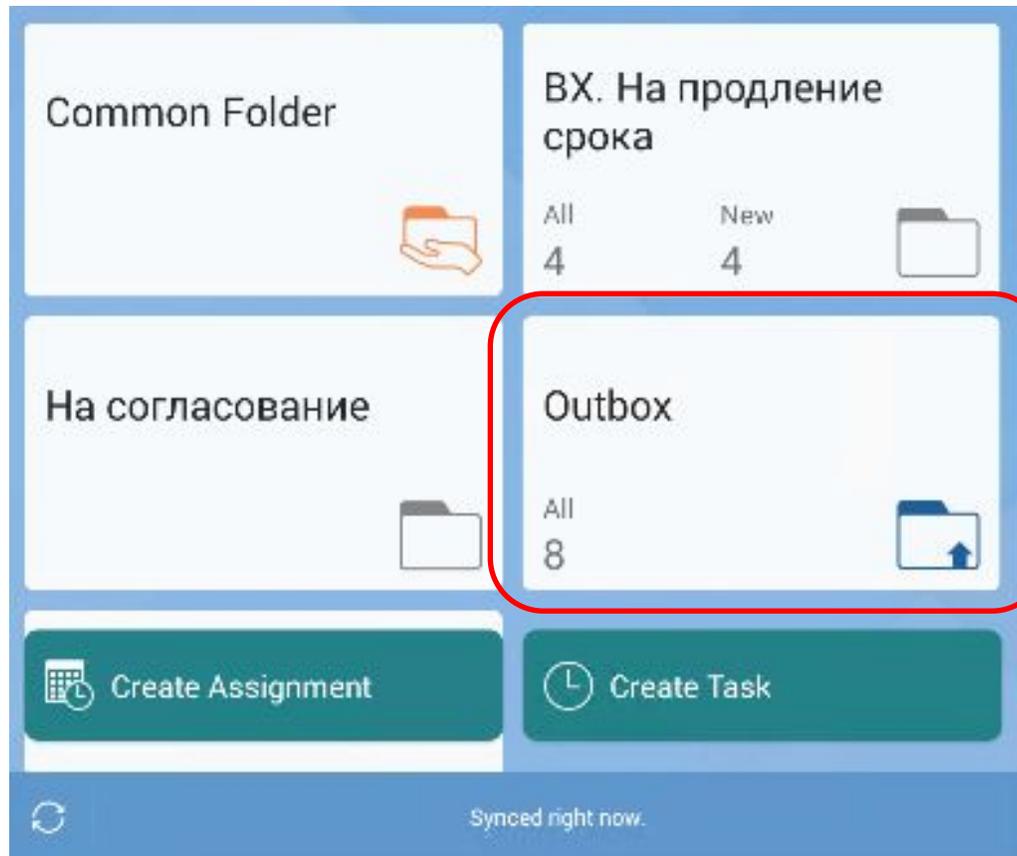
# REPL (*read-eval-print-loop*)

```
>>> tree
[[object CalabashRootView] > PhoneWindow$DecorView]
  [ActionBarOverlayLayout] id: "decor_content_parent"
    [FrameLayout] id: "content"
      [LinearLayout]
        [EditText] id: "creditCardNumberText"
        [Button] id: "validateButton" text: "Validate Credit Card"
        [TextView] id: "errorMessagesText"
    [ActionBarContainer] id: "action_bar_container"
      [ActionBarView] id: "action_bar"
        [LinearLayout]
          [ActionBarView$HomeView]
            [ImageView] id: "home"
            [LinearLayout]
              [TextView] id: "action_bar_title" text: "Enter Credit Card Number"
          ]
        ]
      ]
    ]
  ]
>>>
```

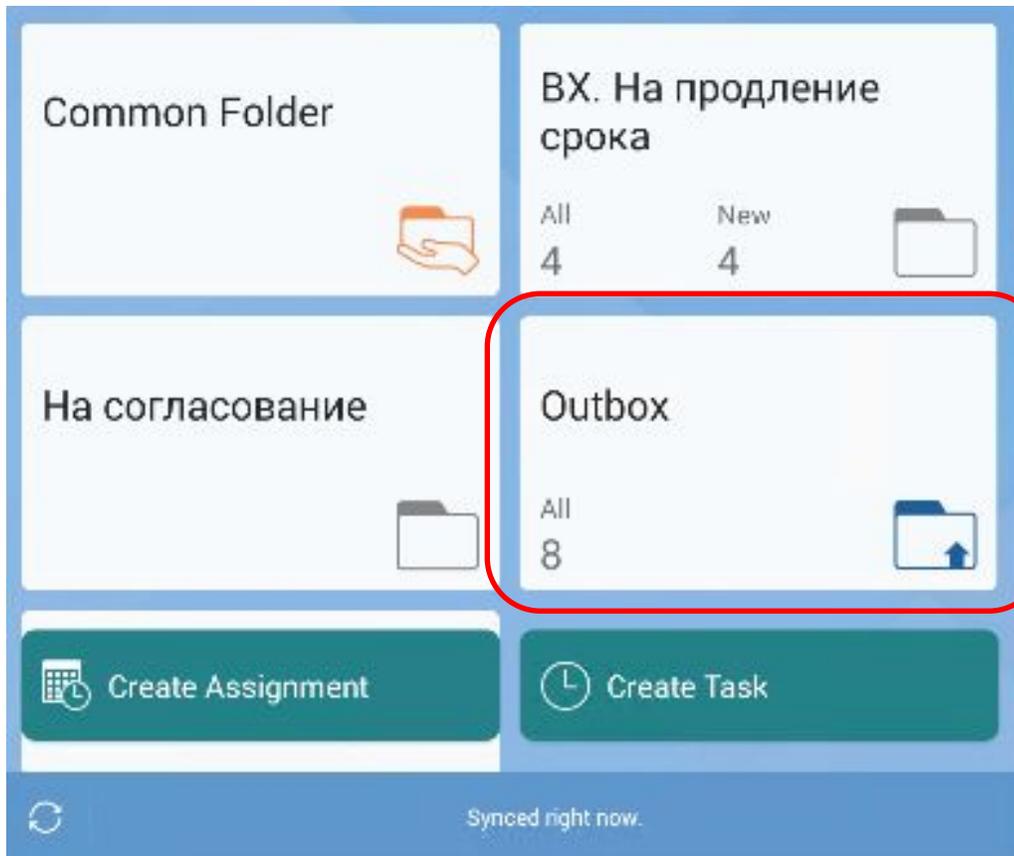


Немного боли

Цель – перейти в папку Outbox и выполнить в ней определенные действия



Цель – перейти в папку Outbox и выполнить в ней определенные действия



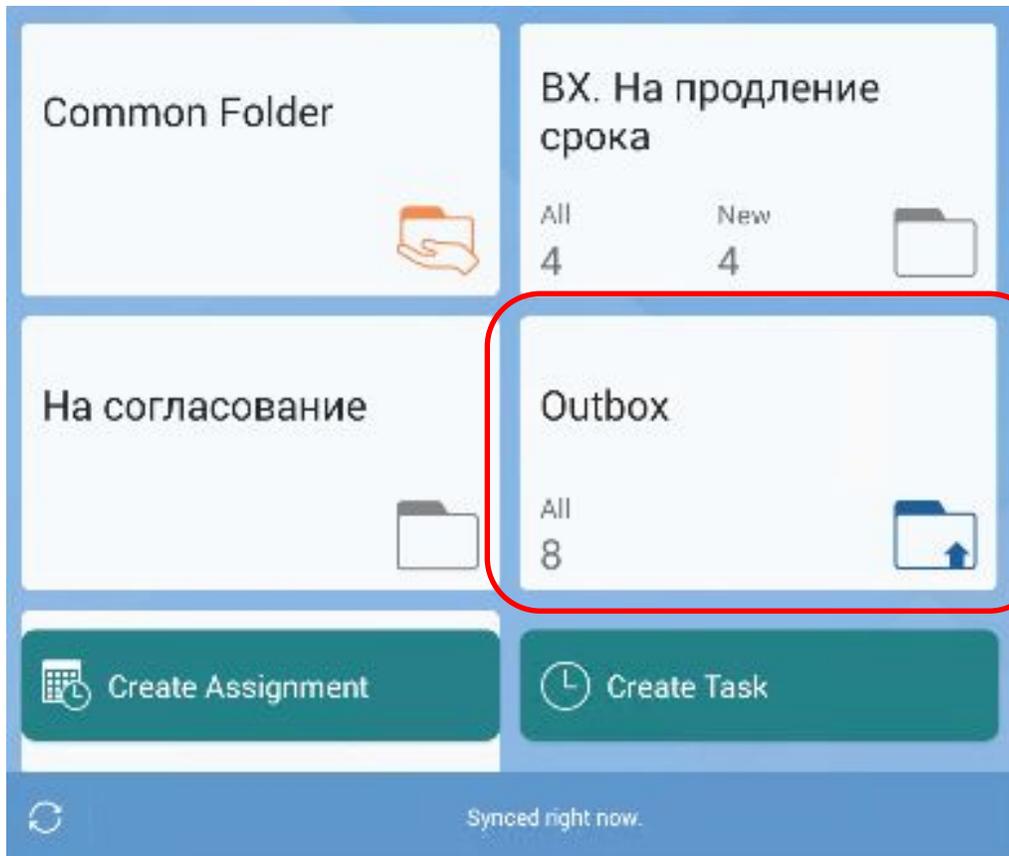
Пишем тест.

...

```
app.OpenFolder(Strings.ExplorerFoldersOutbox);  
app.Tap(Resources.Identifiers.HeaderTaskJob);
```

...

Цель – перейти в папку Outbox и выполнить в ней определенные действия



Пишем тест.

Нужные нам шаги:

```
app.OpenFolder(Strings.ExplorerFoldersOutbox);  
app.Tap(Resources.Identifiers.HeaderTaskJob);
```

Запускаем локально – всё ок.

Запускаем на стенде АТ – падает при открытии папки

Common Folder

ВХ. На продление срока

All 4    New 4

На согласование

Outbox

All 8

Create Assignment    Create Task

Synced right now.

Detailed description: This screenshot shows a mobile application interface with a grid of folder cards. The top row contains 'Common Folder' and 'ВХ. На продление срока'. The 'ВХ. На продление срока' card shows 'All 4' and 'New 4' with a folder icon. The second row contains 'На согласование' and 'Outbox'. The 'Outbox' card shows 'All 8' and a folder icon with an upward arrow. Below the folders are two green buttons: 'Create Assignment' and 'Create Task'. At the bottom, a status bar shows a refresh icon and the text 'Synced right now.'

For Flags

All 7    New 2

Favorites

All 5

Common Folder

ВХ. На продление срока

All 4    New 4

Create Assignment    Create Task

Synced 2 minutes ago.

Detailed description: This screenshot shows a mobile application interface with a grid of folder cards. The top row contains 'For Flags' and 'Favorites'. 'For Flags' shows 'All 7' and 'New 2' with a folder icon. 'Favorites' shows 'All 5' with a folder icon containing a star. The second row contains 'Common Folder' and 'ВХ. На продление срока'. The 'ВХ. На продление срока' card shows 'All 4' and 'New 4' with a folder icon. Below the folders are two green buttons: 'Create Assignment' and 'Create Task'. At the bottom, a status bar shows a refresh icon and the text 'Synced 2 minutes ago.'

# Ограничения Xamarin.UITest

- нет возможности запускать iOS тесты из Visual Studio for Windows;
- нет возможности работать с системными уведомлениями и диалогами;
- нет возможности управлять системными параметрами (сеть, интернет подключение и т.п.);
- нет интеграции с другими приложениями на устройстве.

# Автотесты в CI

- Unit тесты запускаются на PR;
- Integration тесты на ночных сборках сервиса;
- E2E тесты ежедневно по расписанию на dev-ветке.

# Автотесты в CI

- Билд в TFS

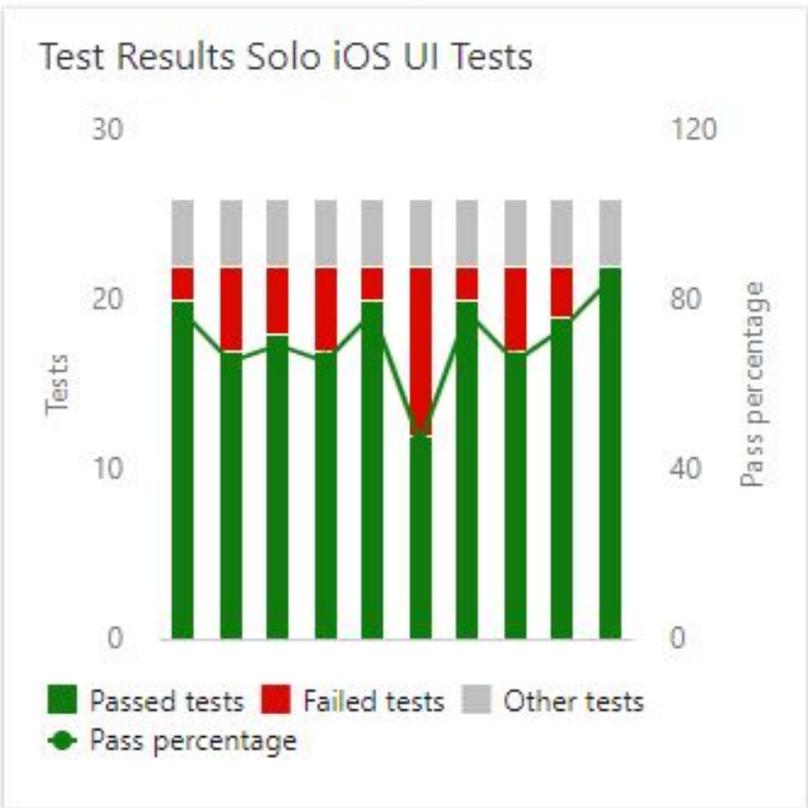
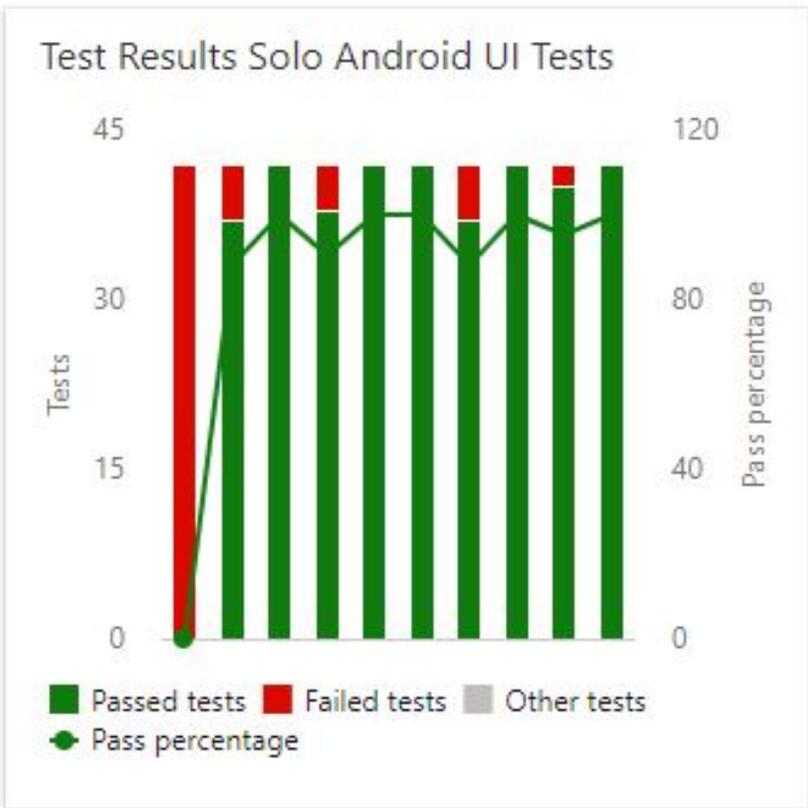
Phase 1  
Run on agent

---

-  Use NuGet 5  
NuGet Tool Installer
-  NuGet restore Cross.sln  
NuGet
-  Set variable DynamicConstants  
Set Variable
-  Build Xamarin.Android Project Solo/Android/Solo.Androi...  
Xamarin.Android
-  Build solution Tests/Solo.Android.UITests/Solo.Android.UI...  
MSBuild
-  Launch MS Emulator  
Run Inline Powershell
-  Launch adb and connect to emulator  
Run Inline Powershell
-  Unlock Screen  
Run Inline Powershell
-  RunTests  
Visual Studio Test
-  Shutdown emulator  
Run Inline Powershell

# Автотесты в CI

Анализ  
результатов  
в TFS



# Итог

- писать автотесты, когда в этом есть смысл:
  - стабильная функциональность;
  - планы развития приложения.

# ИТОГ

- писать автотесты, когда в этом есть смысл;
- решать применением автотестов ваши проблемы;

# Итог

- писать автотесты, когда в этом есть смысл;
- решать применением автотестов ваши проблемы, например:
  - уменьшить количество багов на регрессии;
  - снизить количество ручного тестирования;
  - ускорить релизы;
  - ...

# Итог

- писать автотесты, когда в этом есть смысл;
- решать применением автотестов ваши проблемы;
- количество тестов, конфигураций, подход к написанию и прогону тестов зависят от ваших целей;
- соблюдать принцип пирамиды при автоматизации тестирования

# Спасибо за внимание

